X Collaboration Meeting of the MPD Experiment at the NICA Facility

Machine learning for particle identification

Artem Korobitsin[1]

Alexey Aparin[1], Alexander Mudrokh[1], Vladimir Popoyan[2], Grigorii Tolkachev[3]

[1] LHEP JINR, [2] MLIT JINR, [3] MEPHI

Dubna
10 November 2022

Outline

- Application of Machine Learning (ML) algorithms for particle identification
- ML model: Boosting Decision Trees (CatBoost) and MLP models
- Data and Feature selection
- Training and testing:
  - ML for PID
  - Comparison with n-sigma method
- Conclusion

Particle identification (PID)

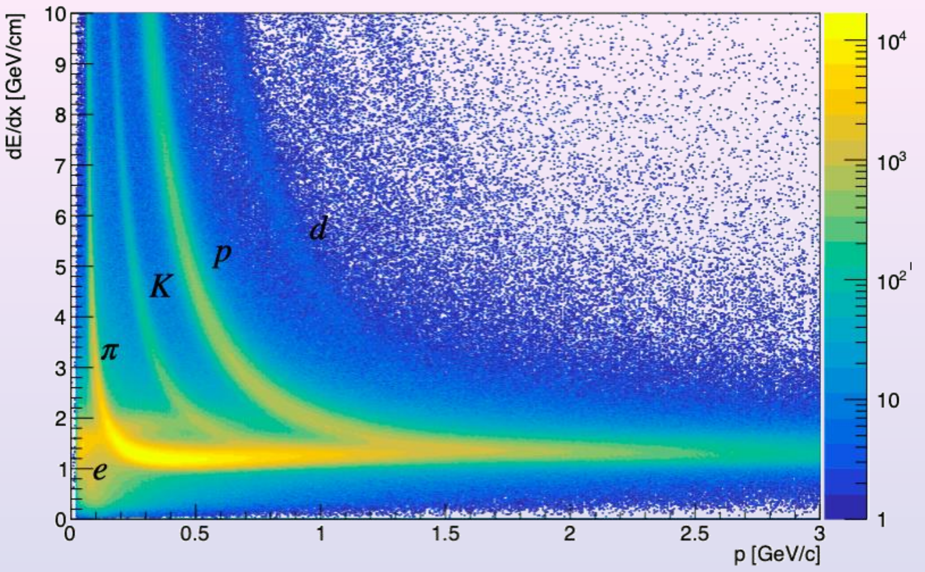
Traditional PID (n-sigma method, Bayesian approach):
– a typical analyzer selects particles “manually” by cutting on certain quantities, like the number of standard deviations of a signal from the expected value (nσ)
– most limitations come in the regions where signals from different particle species cross
– “cut” optimization is a timeconsuming task

Machine learning PID:
- good task for machine learning
- can learn non-trivial relations between different track parameters and PID

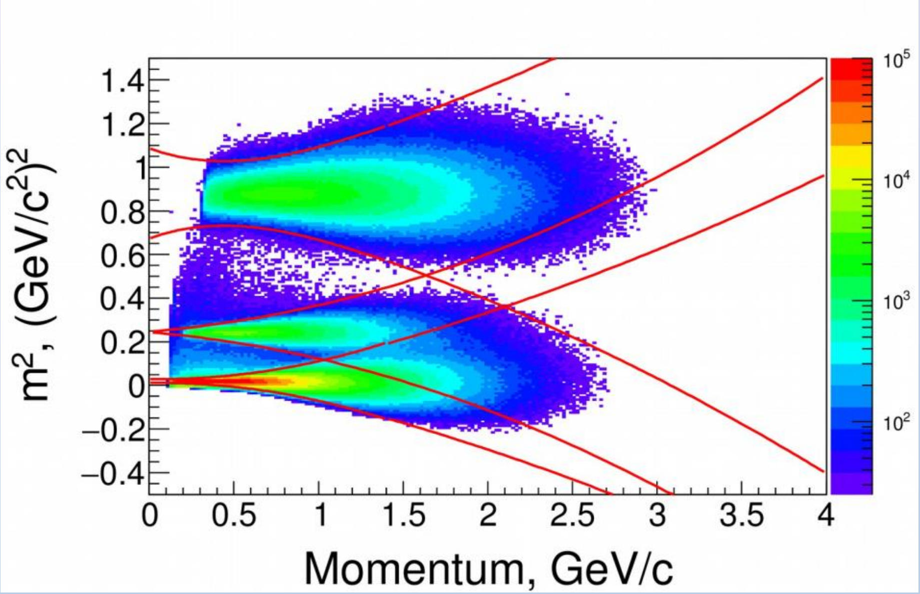
Proposed solution for PID
Build a ML classifier that can outperform traditional PID
Train and validate the classifier on Monte Carlo data

The classifier is a “black box” - no easy way to tell what’s going on inside

dE/dx vs momentum in TPC

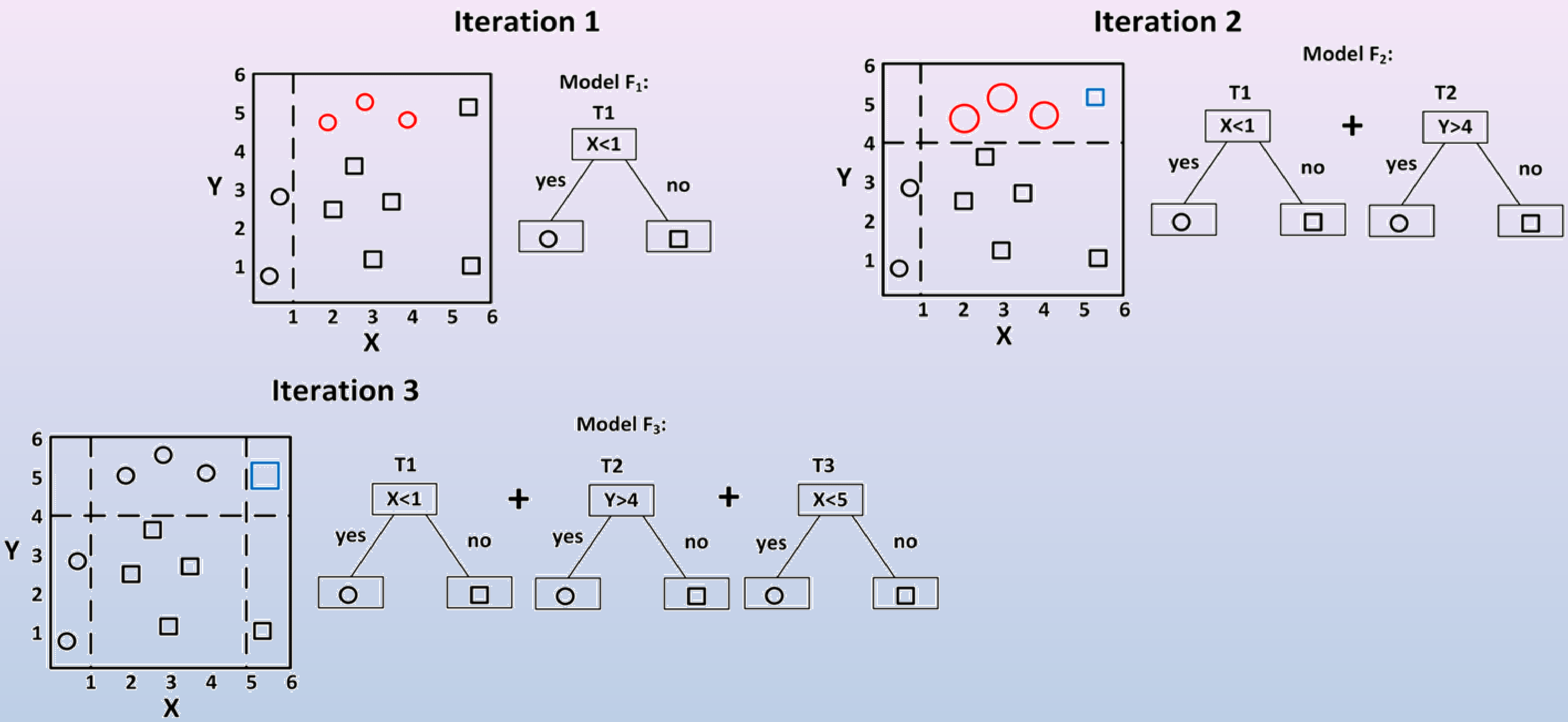


m^2 vs. momentum in TOF



ML Model: Boosting Decision Trees (CatBoost)

is a machine learning algorithm that uses gradient boosting on decision trees.
At each iteration, trees are added in such a way that the value of the objective function decreases.



https://www.researchgate.net/figure/A-simple-example-of-visualizing-gradient-boosting_fig5_326379229

Multi-layer Perceptrons (MLP)

one of the standard method for multi-class and binary classification the evaluation.

TPC

TOF

V_x, V_y, V_z

P

eta

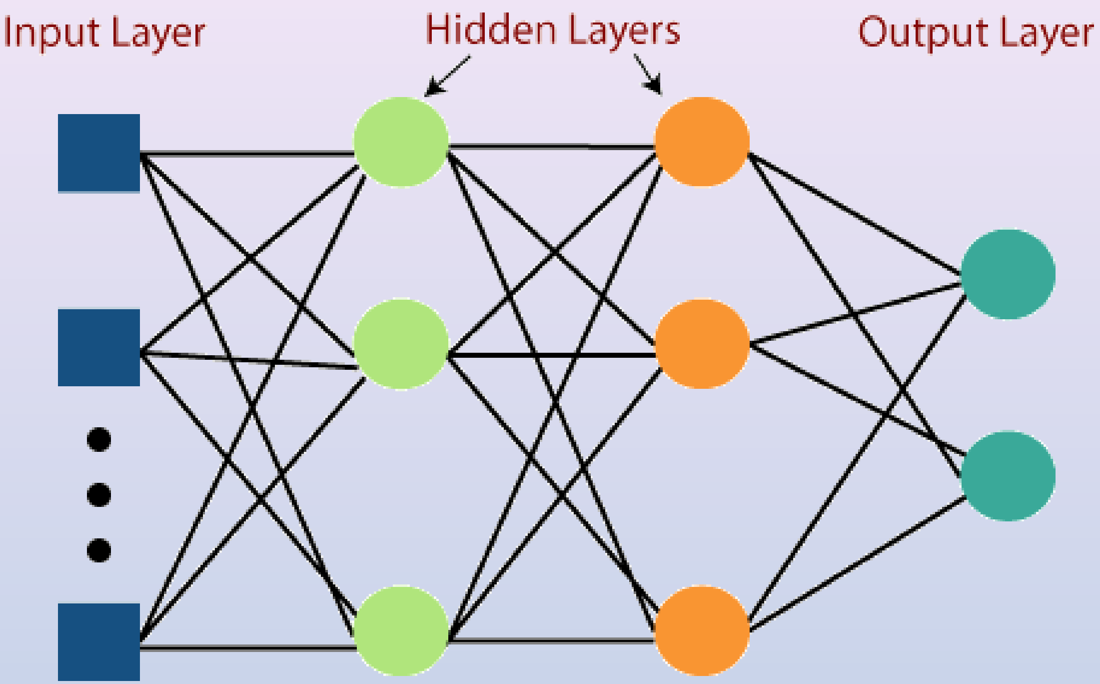
charge

dca

nHits

dE/dx

m^2



π^+

π^-

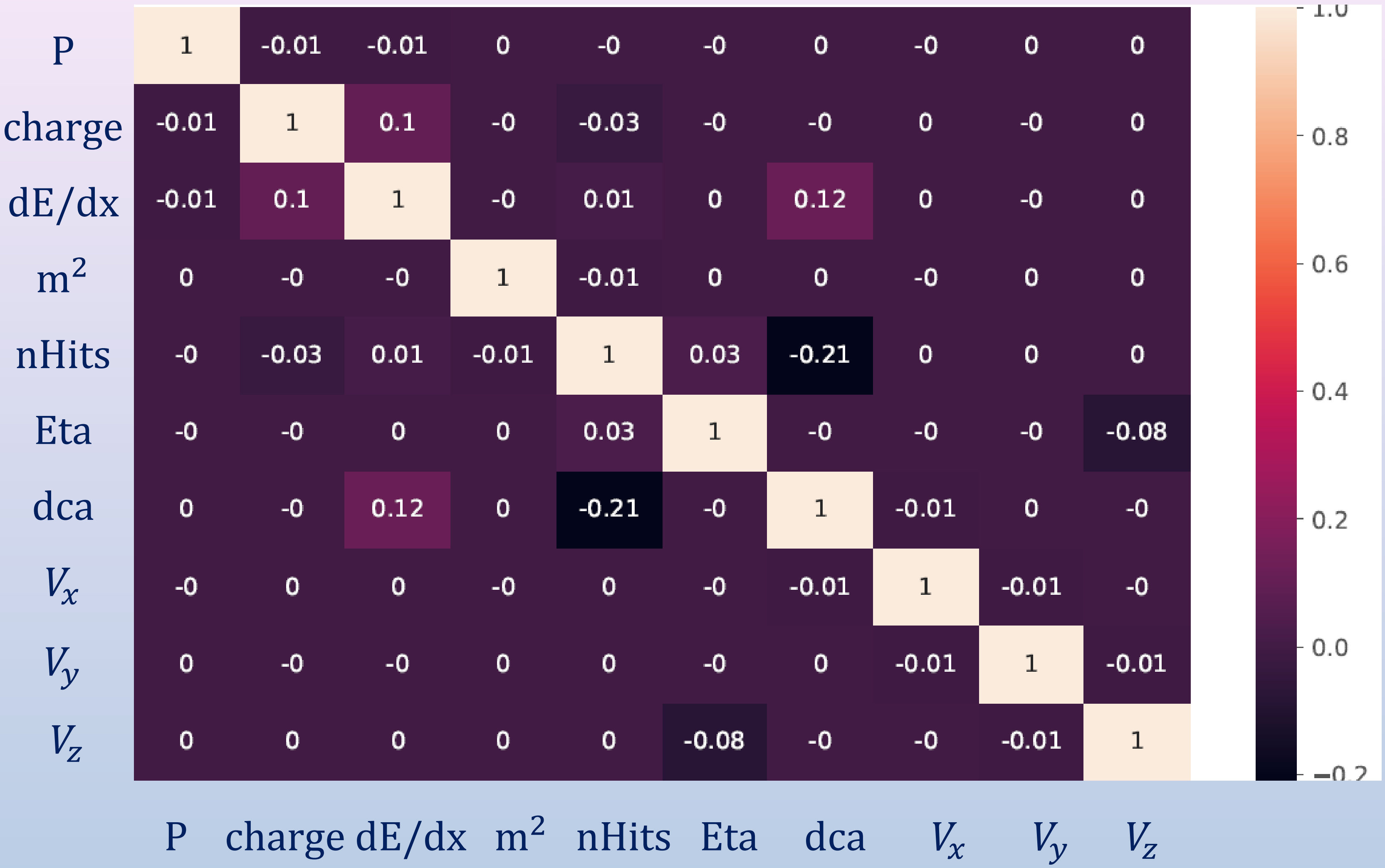
K^+

K^-

p

$\bar{p}$

Correlation matrix for all input feature



Test data sample:

prod1: UrQMD v.3.4 + BOX + Geant-4 based general-purpose simulation project for minbias (b = 0-16 fm) Bi (83/209) +Bi (83/209) collisions at 9.2 GeV, full detector configuration.

prod4: UrQMD v.3.4 + BOX + Geant-4 based general-purpose simulation project for minbias (b = 0-16 fm) Bi (83/209) +Bi (83/209) collisions at 9.2 GeV, full detector configuration. + SmearVertexXY = 1.1 cm

prod05: Request 25 UrQMD + Geant-4 based general-purpose simulation project for minbias (b = 0-16 fm) Bi (83/209) +Bi (83/209) collisions at 9.2 GeV, full detector configuration.

Training and validation dataset:

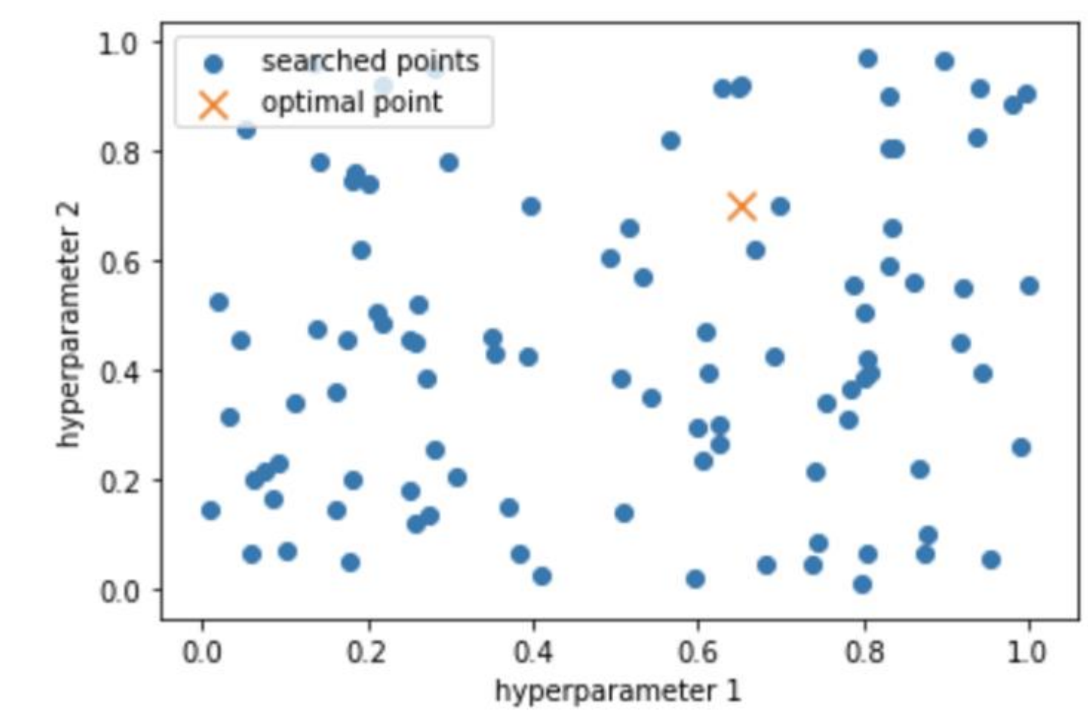
1 million elements (tracks) for each of the six classes (particles): $\pi^+, \pi^-, K^+, K^-, p, \bar{p}$

Testing dataset: 50000 events.

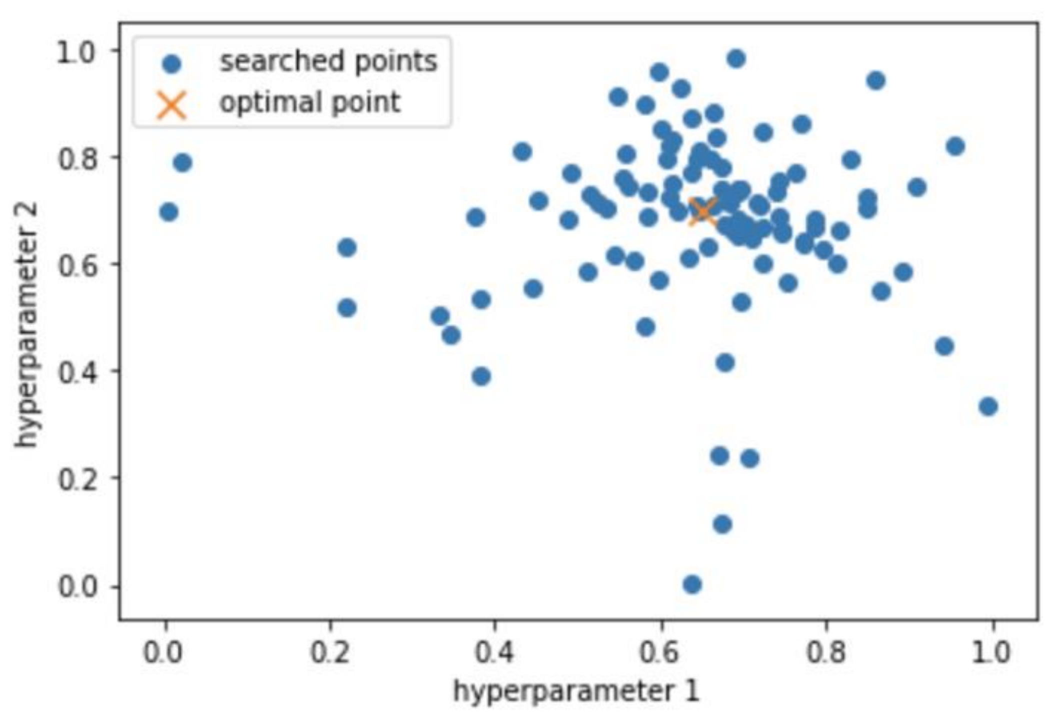
Hyperparameters selection (Select optimal hyperparameters of ML model)

Four commonly used optimization strategies: Grid search, Random search, Hill climbing and Bayesian optimization.

Random search



Bayesian optimization



https://miro.medium.com/max/4800/1*tYWqO5BwNDVaM3kP3w1IAg.png

Feature selection

prod 01:

	Feature Id	Importances
0	charge	48.976478
1	p	15.612522
2	m2	13.219858
3	dedx	12.504383
4	dca	2.931781
5	nHits	2.682914
6	eta	1.732293
7	Vz	0.904500
8	Vx	0.757425
9	Vy	0.677845

prod 04:

	Feature Id	Importances
0	charge	52.595520
1	p	16.143578
2	m2	11.179546
3	dedx	9.959441
4	eta	3.202594
5	dca	3.178775
6	nHits	2.890517
7	Vy	0.322261
8	Vx	0.293670
9	Vz	0.234098

prod 05:

	Feature Id	Importances
0	charge	43.753433
1	p	19.143319
2	dedx	18.371532
3	m2	9.106441
4	dca	3.549774
5	nHits	2.178229
6	eta	1.912249
7	Vz	0.802412
8	Vx	0.630954
9	Vy	0.551657

The bigger the value of the importance the bigger on average is the change to the prediction value, if this feature is changed.

Confusion matrix for the six classes of model

Each column of matrix – predicted value, each row of matrix – true value.

prod 01:

True label \ Predicted label	π^+	k^+	p	π^-	k^-	$\bar{p}$
π^+	99.22%	0.49%	0.27%	0.02%	0.00%	0.00%
k^+	18.32%	79.66%	1.97%	0.04%	0.00%	0.00%
p	6.63%	1.97%	91.37%	0.03%	0.00%	0.00%
π^-	0.02%	0.00%	0.00%	99.50%	0.36%	0.12%
k^-	0.01%	0.01%	0.01%	22.87%	76.58%	0.52%
$\bar{p}$	0.02%	0.00%	0.00%	12.28%	3.27%	84.44%

prod 04:

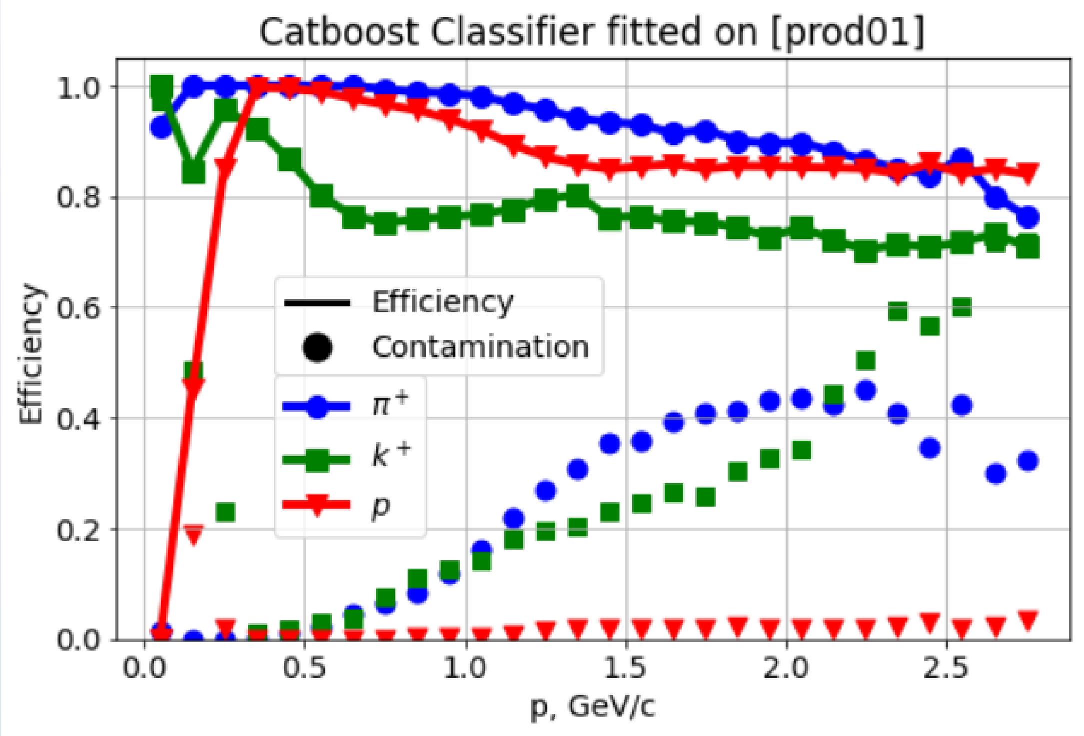
True label \ Predicted label	π^+	k^+	p	π^-	k^-	$\bar{p}$
π^+	98.30%	1.23%	0.44%	0.01%	0.01%	0.02%
k^+	11.76%	85.42%	2.73%	0.00%	0.00%	0.09%
p	3.92%	1.72%	94.29%	0.00%	0.00%	0.06%
π^-	0.01%	0.00%	0.00%	98.58%	0.87%	0.54%
k^-	0.00%	0.01%	0.00%	13.62%	82.78%	3.59%
$\bar{p}$	0.00%	0.00%	0.00%	5.09%	0.84%	94.07%

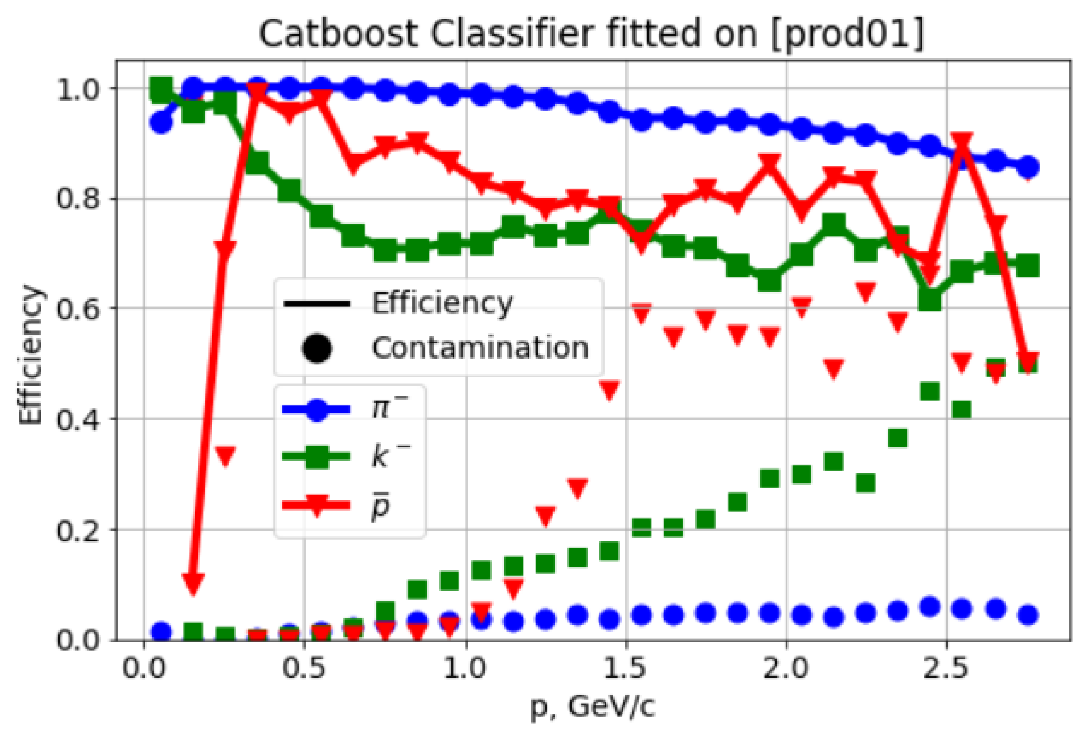
prod 05:

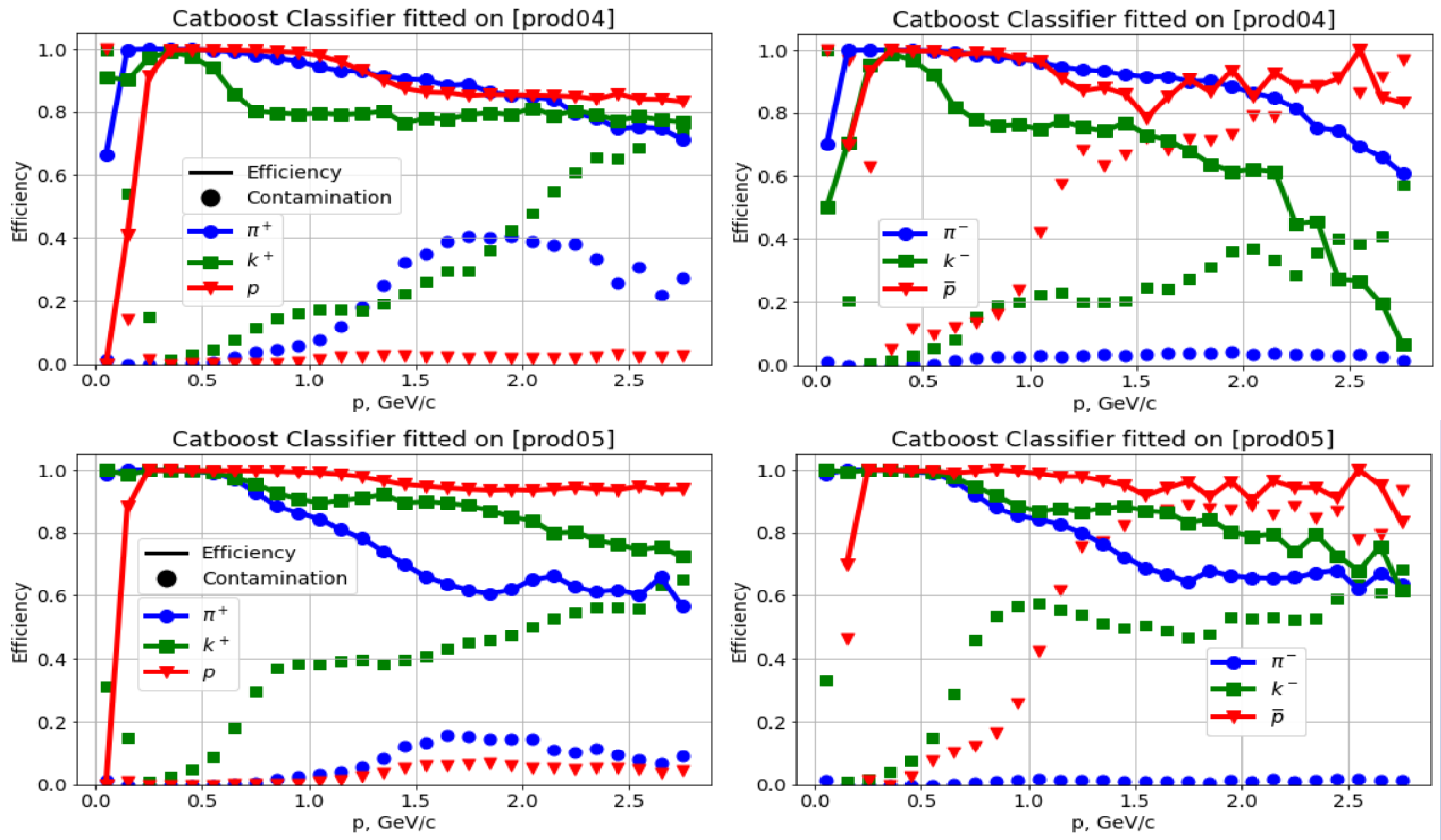
True label \ Predicted label	π^+	k^+	p	π^-	k^-	$\bar{p}$
π^+	95.74%	3.21%	1.03%	0.01%	0.00%	0.00%
k^+	3.85%	93.76%	2.35%	0.00%	0.02%	0.01%
p	0.78%	1.64%	97.55%	0.00%	0.01%	0.02%
π^-	0.01%	0.00%	0.00%	95.81%	3.18%	0.99%
k^-	0.00%	0.01%	0.01%	4.16%	93.88%	1.93%
$\bar{p}$	0.00%	0.02%	0.02%	0.76%	1.46%	97.75%

CatBoost results

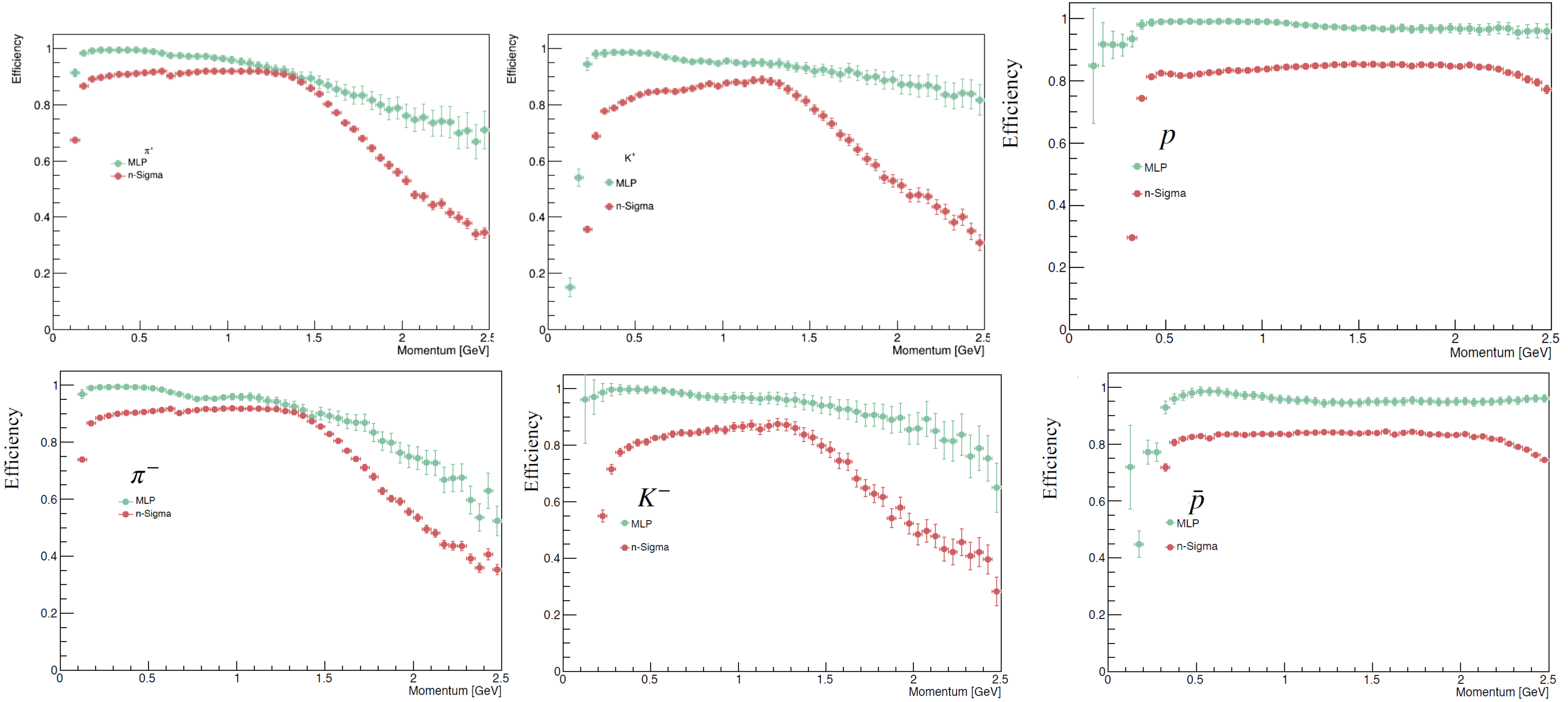
$$Efficiency = \frac{right\ identified\ tracks}{all\ tracks} \qquad Contamination = \frac{wrong\ identified\ tracks}{identified\ tracks}$$





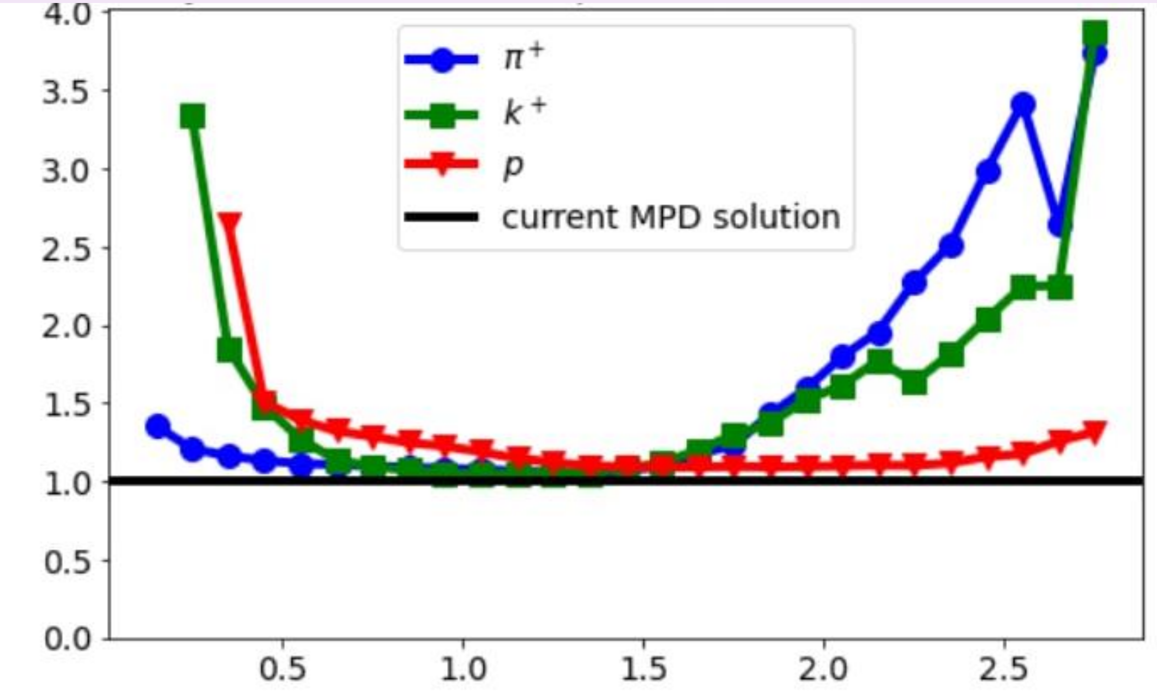

Catboost Classifier fitted on [prod04]
Efficiency
Contamination
π^+
k^+
p
Efficiency
p, GeV/c
Catboost Classifier fitted on [prod04]
π^-
k^-
$\bar{p}$
Efficiency
p, GeV/c
Catboost Classifier fitted on [prod05]
Efficiency
Contamination
π^+
k^+
p
Efficiency
p, GeV/c
Catboost Classifier fitted on [prod05]
π^-
k^-
$\bar{p}$
Efficiency
p, GeV/c

Comparison MLP with n-sigma method

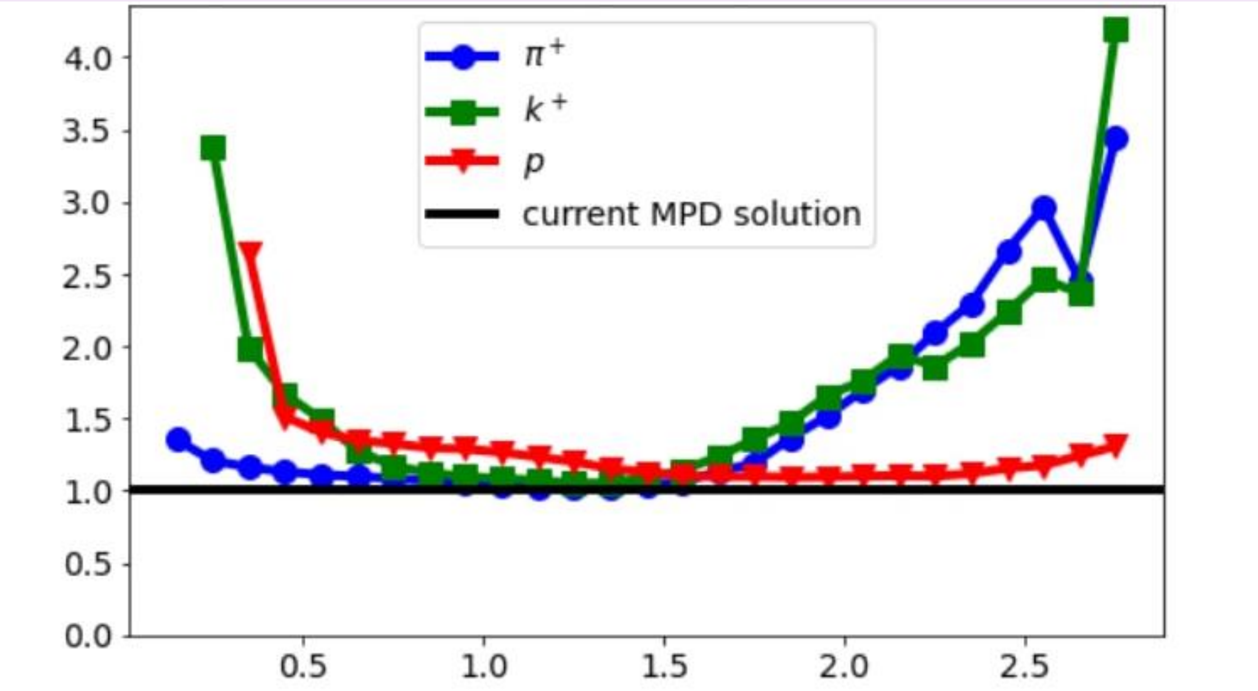


Efficiency ratio of CatBoost and n-sigma method

prod01



prod04



Conclusion

ML-based PID outperforms traditional PID, especially in the low and high momentum region.

Training needed only once for each data set – no need for manual cut optimizations.

Shown improvement only for the several datasets of MC simulation data. Planned to conduct research for a wide set of MC data.

Thank you for your attention